AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph beginning on page 1 titled "RELATED APPLICATIONS", as follows:

This application is related to concurrently filed and commonly assigned U.S. Patent Application Serial Number ~~09/896,019~~ 09/896,019 entitled "SYSTEM AND METHOD FOR FILE SYSTEM MANDATORY ACCESS CONTROL," the disclosure of which is hereby incorporated herein by reference.

Please amend the paragraph beginning on page 15 as follows:

Referring now to Fig. 4, there is illustrated an exemplary architecture of a trusted Linux host OS, which implements compartments to provide containment. System 400 includes a plurality of compartments. In this example, WEB compartment 401, FTP compartment 402, and SYSTEM compartment 403 are shown. Each compartment may be associated with various executing processes or threads. Thus, with reference to Fig. 4, a base Linux kernel 400 generally comprises TCP/IP Networking means 406, UNIX domain sockets 408, inter-process communication means 410 (e.g., a Sys V IPC means), file access module 412, and other subsystems 408. The trusted Linux OS additionally comprises kernel extensions 415 in the form of a security module 421, a device configuration module 418, a rule database 416 and kernel modules 422. As shown, at least some of the Linux kernel subsystems 406, 408, 410, 412, and 414 have been modified to make call outs to the kernel-level security module 421. In this exemplary implementation, the security module 421 makes access control decisions and is responsible for enforcing the concept of a compartment, thereby providing containment. Such exemplary OS architecture in which embodiments of the present invention may be implemented is further described in concurrently filed and commonly assigned U.S. Patent Application Serial Number ~~09/896,019~~ 09/896,019 entitled "SYSTEM AND METHOD FOR FILE SYSTEM MANDATORY ACCESS CONTROL," the disclosure of which has been incorporated herein by reference.

Please amend the paragraph beginning on page 15, line 17 as follows:

Referring now to Fig. 4, there is illustrated an exemplary architecture of a trusted Linux host OS, which implements compartments to provide containment. System 400 includes a plurality of compartments. In this example, WEB compartment 401, FTP compartment 402, and SYSTEM compartment 403 are shown. Each compartment may be associated with various executing processes or threads. Thus, with reference to Fig. 4, a base Linux kernel 400 generally comprises TCP/IP Networking means 406, UNIX domain sockets 408, inter-process communication means 410 (e.g., a Sys V IPC means), file access module 412, and other subsystems ~~408~~ 414. The trusted Linux OS additionally comprises kernel extensions ~~415~~ in the form of a security module 421, a device configuration module 418, a rule database 416 and kernel modules 422. As shown, at least some of the Linux kernel subsystems 406, 408, 410, 412, and 414 have been modified to make call outs to the kernel-level security module 421. In this exemplary implementation, the security module 421 makes access control decisions and is responsible for enforcing the concept of a compartment, thereby providing containment. Such exemplary OS architecture in which embodiments of the present invention may be implemented is further described in concurrently filed and commonly assigned U.S. Patent Application Serial Number ~~_____~~ 09/896,019 entitled "SYSTEM AND METHOD FOR FILE SYSTEM MANDATORY ACCESS CONTROL," the disclosure of which has been incorporated herein by reference.

Please amend the paragraph beginning on page 16, line 7 as follows:

Security module 421 additionally consults rule database 416 when making a decision. Rule database 416 contains information about allowable communication paths between compartments, thereby providing narrow, well-controlled interfaces into and out of a compartment. Thus, the processes of the compartments are limited to accessing system resources according to the rules stored in rule database 416. Rule database 416 may comprise separate tables for TCP/IP networking resource rules and for file system resource rules. Also, the various components can be stored in different locations. For example, TCP/IP resource rules may be stored in random access memory, while file system resource rules may be stored on the file system. Fig. 4 also illustrates how the kernel extensions ~~415~~ are administered from user space 420 via a series of custom system calls. As described further below, such

custom system calls may include: some to manipulate the rule table 416 and others to run processes in particular compartments and configure network interfaces.

Please amend the paragraph beginning on page 32, line 22 as follows:

According to at least one embodiment of the present invention, "tlrules" utilities are provided for manipulating compartment rules. Such "tlrules" utilities may comprise command-line utilities for adding, deleting and listing rules. In the exemplary implementation described above with Fig. 4, such tlrules may be implemented as command-line utilities ~~440~~ 404 that are executable to manipulate compartment rules (e.g., within rule database 416) via /proc/tlx interface provided by a kernel-loadable module. Rules can either be entered on the command line, or can be read from a text file.

In accordance with the exemplary implementation described above, rules may take the following format:

<rule>::=<source>[<port>]-><destination>[<port>]<method list><netdev>

where:

<identifier>== (<compartment> _ <host> _ <net>) [<port>]

<compartment>== "COMPARTMENT" <comp_name>

<host>=="HOST"<host_name>

<net>=="NET"<ip_addr> <netmask>

<net>=="NET"<ip_addr> ?/? <bits>

<comp_name>== A valid name of a compartment

<host_name>== A known hostname or IP address

<ip_addr>== An IP address in the form a.b.c.d

<netmask>== A valid netmask, in the form a.b.c.d

<bits>== The number of leftmost bits in the netmask.... 0 thru 31

<method_list>== A list of comma-separated methods (In this exemplary embodiment, methods supported are: TCP (Transmission Control Protocol), UDP (User Datagram Protocol), and ALL.